

22141 U.S. PTO  
10/761377  
  
012204

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-172521

(43)Date of publication of application : 23.06.2000

(51)Int.CI.

G06F 11/18

G06F 15/177

(21)Application number : 10-351792

(71)Applicant : TOYOTA MOTOR CORP

(22)Date of filing : 10.12.1998

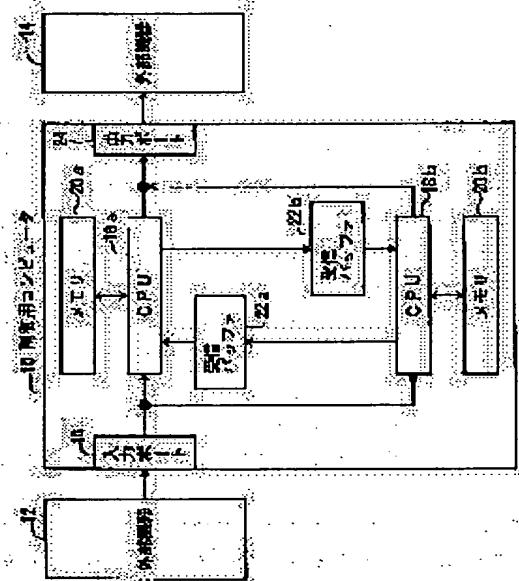
(72)Inventor : UCHIDA KIYOKI

### (54) ABNORMALITY DETECTING METHOD FOR CPU

#### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a method for easily detecting the abnormality of a CPU without synchronizing a plurality of CPUs with each other.

**SOLUTION:** An input signal is supplied from an external device 12 to CPUs 18a and 18b through an input port 16. The CPUs 18a and 18b obtain the latent arithmetic values of the opposite CPUs through receiving buffers 22a and 22b and compares the arithmetic values with the latest three arithmetic values. Then the CPUs 18a and 18b judge that at least one of the CPUs 18a and 18b in a control computer 10 is abnormal when the latest arithmetic values of the opposite CPUs are larger than the maximum values among their arithmetic values or smaller than the minimum values. If the CPU 18a is judged to be abnormal, the output of a control signal supplied to the external device 14 is stopped. When it is judged that the CPU 18b is abnormal, on the other hand, the output of a control signal from the CPU 18a is inhibited.



#### LEGAL STATUS

[Date of request for examination] 29.08.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

**THIS PAGE BLANK (USPTO)**



## なCPUの異常検出が実現する

[00101] また、上記目的は、前記項2に記載する如く、請求項1記載のCPUの異常検出方法であって、前記第4のステップにおける最新の所定数の演算値は、最新の3つの演算値であるCPUの異常検出方法により達成される。各CPUが他のCPUが算出するたままでに、所定の最小の時間を要するため、各CPUが算出した他のCPUの異常検出方法は、その前の回数の演算処理時から最新の演算処理時に間に算出されたものであると認定できる。本発明によると、相手側CPUの最新の演算値と比較する自己CPUの演算値が最新の3つの演算値に限定されるので、比較時の誤差の許容範囲をより小さくして設定することができる。誤差範囲を小さくして設定することで、CPUの異常の検出がより高精度に行なわれる。

[0011]

【発明の実施の形態】以下、図1～図4を用いて本発明の実施の形態について説明する。図1は、本発明の異常検出方法によってCPUの異常を検出する制御用コンピュータ100の構成図である。この制御用コンピュータ100は、外部機器1から順次与えられる入力信号に基づいて演算処理を行う。そして、制御用コンピュータ100は、演算処理の結果に応じた制御信号を外部機器1に与えて外部機器1の動作制御を行なう。制御用コンピュータ100に入力信号を供給する外部機器1は、例えば、スイッチやセンサ等である。また、制御用コンピュータ100によって制御される外部機器1は、例えば、アクチュエータやLED表示器等である。

[0012] 図1に示すように、制御コンピュータ100は、入力ポート16、CPU18a、18b、メモリ20、20b、受信バッファ22a、22b及び出力ポート40、40b等を有する。入力ポート16は、スイッチやセンサ等で構成される外部機器1から入力信号を読み込む。そして、入力ポート16は、ノイズ除去処理やレベルシフト処理等を施した後の入力信号をCPU18a、18bに与える。

[0013] CPU18aは、外部機器1から入力ポート16を介して与えられた入力信号に基づき演算処理を行なって、入力信号に応じた演算値a1、a2、a3、...を算出する。同様にCPU18bも外部機器1から入力ポート16を介して与えられた入力信号に基づき演算処理を行なって、入力信号に応じた演算値b1、b2、b3、...を算出する。

おらず、共に新定の間隔t(例えば、約6ms)で演算値の算出を繰り返す。従って、演算値a1、a2、a3、...と演算値b1、b2、b3、...は、例えば、b1、a1、b2、a2、b3、a3、...の順で交互に算出される。CPU18aによって算出された演算値a1、a2、a3、...は、メモリ20cに記録するまでは、所定の最小の時間を要するため、各CPUが他のCPUの異常検出方法は、その前の回数の演算処理時から最新の演算処理時に間に算出されたものであると認定できる。本発明によると、相手側CPUの最新の演算値と比較する自己CPUの演算値が最新の3つの演算値に限定されるので、比較時の誤差の許容範囲をより小さくして設定することができる。誤差範囲を小さくして設定することで、CPUの異常の検出がより高精度に行なわれる。

など漫信ハッフル226に載次掲載される。また、CP

U18bによって算出された演算値b1、b2、b3、  
 ・・・は、メモリ20bと受信バッファ22aに履帯格納される。  
 [0015]また、CPU18aは、最新の3つの自己の演算値（例えば、a1、a2、a3）をメモリ20aから読み出し、最新のCPU18bの演算値（例えば、b3）を受信バッファ22aから読み出す。そして、CPU18aは、演算値a1、a2、a3と演算値b3を比較し、比較結果に応じた制御信号を出力ポート24を介してアクチュエータやLED表示器等で構成される外部機器14に与える。この時、外部機器14は、CPU18aから与えられた制御信号に従って動作する。  
 [0016]一方、CPU18bは、最新の3つの自己の演算値（例えば、b1、b2、b3）をメモリ20bから読み出し、最新のCPU18aの演算値（例えば、a3）を受信バッファ22bから読み出す。そして、CPU18bは、演算値b1、b2、b3と演算値a3を比較し、比較の結果、必要に応じてCPU18aから外部機器14への制御信号の出力を禁止にする。  
 [0017]メモリ20a、20bは、それぞれCPU18a、18bの演算処理の結果である演算値a1、a2、・・・、b1、b2、・・・の他、CPU18a、18bの動作プログラム等を格納する。次に、CPU18a、18bの動作説明をフローチャートを用いて行なう。図2は、CPU18aが実行するルーチンを示すフローチャートである。起始で示すルーチンは、その処理が終了する迄繰り返す。なお、メモリ20aには、前々回と前回のルーチンで取得された入力信号に基づくCPU18aの演算処理の結果である演算値a1、b1、b2が既に格納されているものとする。また、受信バッファ22aには、CPU18bの演算処理の結果である最新の演算値b3が既に格納されているものとする。  
 [0018]図2に示すルーチンが起動されると、まず、ステップ1.0において、外部機器12からの入力信号が入力ポート16を介して取得される。このステップ1.0の処理が終了すると、次に、ステップ1.0の処理が実行される。ステップ1.0では、CPU18bによる演算処理の結果である演算値b3が受信バッファ22aから取得される。そして、次に、ステップ1.0の処理が実行される。  
 [0019]ステップ1.0では、ステップ1.0において取得された入力信号に基づいた所定の演算が実行され、その結果、演算値b3が算出される。そして、既くステップ1.0では、ステップ1.0で算出された演算値a3がメモリ20aに格納される。また、演算値a3は、受信バッファ22bにも与えられ、受信バッファ22b内に格納される。このステップ1.0の処理が終了すると、次に、ステップ1.0の処理が実行される。

0002018032001

ノルマニヤの政治

四百四十一

対応する時刻 t<sub>1</sub>、t<sub>2</sub>、t<sub>3</sub>、t<sub>4</sub>は、各演算値が CPU 18a の最新の演算値 a<sub>3</sub>より先に算出された値のか後に算出された値のものが判別できない。そのため、CPU 18a の前回の演算処理で算出された演算値 a<sub>2</sub>と、今回(最新)の演算値 a<sub>3</sub>と、演算値 a<sub>3</sub>の次に算出される演算値 a<sub>4</sub>などを考慮して、上記ステップ1.8の比較処理における既存等の演算値を大きくして設定する必要がある。

【0030】しかし、本発明では、上記ステップ1.0、2、1.4及びステップ2.0.2、2.0.4に示すように、CPU 18a、18bは、共に相手側 CPU の最新の演算値を受信するバッファ2.2a、2.2bを介して取扱してから、その演算処理を行なっている。そのため、相手側 CPU の最新の演算値は、常に自己 CPU の最新の演算値よりも後に算出されたものであると確定できる。

【0031】また、相手側 CPU が最新の演算値を算出した後に、自己 CPU がその演算値を受信するバッファを介して取扱するまでは所定の最も小さな時刻 t<sub>4</sub>を要するので、本発明において、自己 CPU が受信バッファを介して取扱した相手側 CPU の最新の演算値は、自己 CPU の前々回の演算処理時から今回(最新)の演算処理時の間に算出されたものであると確定できる。

【0032】そこで、本発明のステップ1.0.8、2.0.8では、相手側 CPU の最新の演算値と、自己 CPU の前回、前回及び今回(最新)の演算処理時の演算値である最新の3つの演算値を比較する構成にしている。このように、本発明では、相手側 CPU の最新の演算値と比較する自己 CPU の演算値を最小の3つに限定するので、比較処理における既存等の評価値を定める判定余裕値  $\alpha$ 、 $\beta$ をより小さな値に設定することができる。判定余裕値  $\alpha$ 、 $\beta$ をより小さな値に設定することで、CPU 18a、18bの異常の検出がより高精度に行なわれる。

【0033】ここで、CPU 18a、18bの演算値の比較処理が不必要な入力信号に対しては、CPU 18a、18bが個別に演算処理を行なうようにしてよい。演算値の比較処理が不要な入力信号をCPU 18a、18bのいずれか一方に分離することと、2つのCPU 18a、18bを効率的に利用することができる。なお、上記実施例は、CPU 18a、18bが共にそれぞれのルーチンに従つて動作し、受信バッファがあるが、例えば、CPU 18aをマスター CPU とし、CPU 18bをスレーブ CPU としてもよい。この場合、マスター CPU である CPU 18aがスレーブ CPU である CPU 18bを起動させる。そして、CPU 18aによって起動したCPU 18bが送信バッファに格納していた最新の演算値をCPU 18aに与える。

【0034】図4は、CPU 18a、18bが共に正常である時の演算値 a<sub>1</sub>、a<sub>2</sub>、a<sub>3</sub>及び b<sub>1</sub>を示す図である。なお、演算値 a<sub>1</sub>、a<sub>2</sub>、a<sub>3</sub>及び b<sub>1</sub>に示すCPUを有するコンピュータシステムにおいて、発明の効果、上記の如く、請求項1記載の発明によれば、複数のCPUを有するコンピュータシステムにおいて、CPU 18aからCPU 18bへの演算値の出力を禁止する。

【0035】なお、制御用コンピュータ1.0内のCPUの数は2個に限らず、制御用コンピュータ1.0内に3個以上のCPUを駆動して、互いの演算値の比較によってCPUの異常検出を行なうようにしてよい。上記実施例において、図2のステップ1.0及び図3のステップ2.0の処理が特許請求の範囲に記載の第1のステップに相当し、図2のステップ1.0.2及び図3のステップ2.0.2の処理が特許請求の範囲に記載の第2のステップに相当し、図2のステップ1.0.4及び図3のステップ2.0.4の処理が特許請求の範囲に記載の第3のステップに相当し、図2のステップ1.0.8及び図3のステップ2.0.8の処理が特許請求の範囲に記載の第4のステップに相当し、図2のステップ1.1.0及び図3のステップ2.1.0の処理が特許請求の範囲に記載の第5のステップに相当する。

【0039】

【発明の効果】上記の如く、請求項1記載の発明によれば、複数のCPUを有するコンピュータシステムにおいて、CPU 18aからCPU 18bへの演算値の出力を禁止する。

CPU 18a、18bによって算出された時刻を示す。

【0036】図5は、CPU 18a、18bの少なくとも一方が異常で検出される。また、CPUの異常検出を高精度に行なうことができる。また、請求項2記載の発明によれば、CPUの異常検出をより高精度に行なうことができる。

【図面の簡単な説明】

【図1】本発明の異常検出方法が適用される制御用コンピュータの構成図である。

【図2】CPUが実行するルーチンを示すフローチャートである。

【図3】CPUが実行するルーチンを示すフローチャートである。

【図4】2つのCPUが共に正常である時の演算値の値を示す図である。

CPU 18a、18bによって算出された最新の演算値 a<sub>3</sub>より先に算出された値のか後に算出された値のものが判別できない。そのため、CPU 18aの前回の演算処理で算出された演算値 a<sub>2</sub>と、今回(最新)の演算値 a<sub>3</sub>と、演算値 a<sub>3</sub>の次に算出される演算値 a<sub>4</sub>などを考慮して、上記ステップ1.8の比較余裕を大きくして設定する必要がある。

【0037】図5に示すように、CPU 18bによって算出された最新の演算値 b<sub>3</sub>の値Cは、CPU 18aによって算出された最新の演算値 a<sub>1</sub>の値Dによって時刻 t<sub>1</sub>に算出された演算値 a<sub>1</sub>の値Dよりも大きくなり、時刻 t<sub>4</sub>に算出された演算値 a<sub>3</sub>の値Dよりも大きい。従って、この場合、図2に示したルーチンがCPU 18a、18bによって算出された演算値 a<sub>1</sub>、a<sub>2</sub>、b<sub>3</sub>、a<sub>3</sub>に対して実行される。また、演算値 a<sub>1</sub>、a<sub>2</sub>、b<sub>3</sub>、a<sub>3</sub>に対して実行されるルーチンがCPU 18a、18bによって算出された演算値 a<sub>1</sub>、a<sub>2</sub>、b<sub>3</sub>、a<sub>3</sub>に対して実行されるルーチンと比較して、CPU 18a、18bの少くとも一方が異常であると判断して、外部機器 1.4を制御するための制御信号の出力を停止する。同時に、CPU 18bがCPU 18a、18bの少くとも一方が異常であると判断する場合、CPU 18aから外部機器 1.4への制御信号の出力を禁止する。

【0038】なお、制御用コンピュータ1.0内のCPUの数は2個に限らず、制御用コンピュータ1.0内に3個以上のCPUを駆動して、互いの演算値の比較によってCPUの異常検出を行なうようにしてよい。上記実施例において、図2のステップ1.0及び図3のステップ2.0の処理が特許請求の範囲に記載の第1のステップに相当し、図2のステップ1.0.2及び図3のステップ2.0.2の処理が特許請求の範囲に記載の第2のステップに相当し、図2のステップ1.0.4及び図3のステップ2.0.4の処理が特許請求の範囲に記載の第3のステップに相当し、図2のステップ1.0.8及び図3のステップ2.0.8の処理が特許請求の範囲に記載の第4のステップに相当し、図2のステップ1.1.0及び図3のステップ2.1.0の処理が特許請求の範囲に記載の第5のステップに相当する。

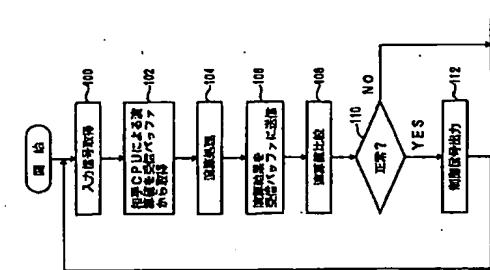
【図1】2つのCPUのうちの少くとも一方が異常で検出される。また、CPUの異常検出を高精度に行なうことができる。また、請求項2記載の発明によれば、CPUの異常検出をより高精度に行なうことができる。

【図2】2つのCPUのうちの少くとも一方が異常で検出される。また、CPUの異常検出を高精度に行なうことができる。

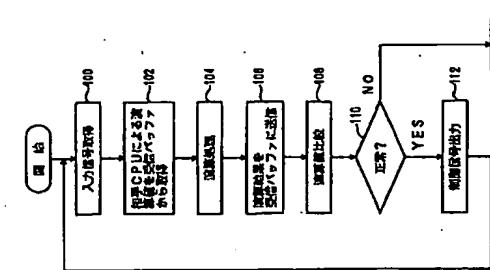
【図3】CPUが実行するルーチンを示すフローチャートである。

【図4】2つのCPUが共に正常である時の演算値の値を示す図である。

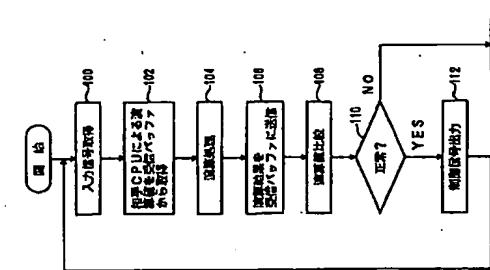
【図1】



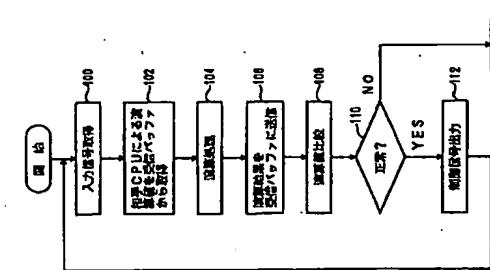
【図2】



【図3】



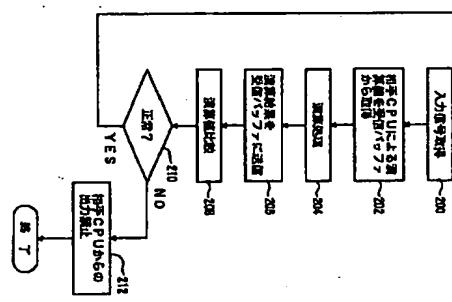
【図4】



(7)

特許平12-172521

[図3]



[図5]

